CLAIMS

The present invention is applicable to systems to process load flow computation by means of parallel algorithm using invented (Fig.4)/available parallel computer. The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. A method of controlling voltages and power flows in a power network, comprising the steps of:

   obtaining on-line data of open/close status of switches and circuit breakers in the power network,

   obtaining on-line readings of real and reactive power assignments or settings at PQ-nodes, real power and voltage magnitude assignments or settings at PV-nodes and transformer turns ratios, which are the controlled variables/parameters,

   performing load-flow computation to calculate complex voltages or voltage magnitude corrections and voltage angle corrections at the power network nodes providing for the calculation of power flowing through different network components, and reactive power generation and transformer tap-position indications,

   evaluating the computed load flow for any of the over loaded power network components and for under/over voltage at any of the nodes,

   correcting one or more controlled parameters and repeating the computing and evaluating steps until evaluating step finds a good power system without any over loaded components and without any under/over voltages in the power network, and

   effecting a change in the power flowing through network components and voltages and phases at the nodes of the power network by actually implementing the finally obtained values of controlled parameters after evaluating step finds a good power system.

2. A loadflow computation defined in claim-1 is characterized in using self-iteration over a node voltage calculation within a network-wide global iteration as depicted by the following iteration-(r+1) equation

$$(V_p^{(sr+1)})^{(r+1)} = [\{(PSH_p - jQSH_p)/((V_p^*)^{sr})^r\} - \sum_{q=1}^{p-1} Y_{pq} V_q^{(r+1)} - \sum_{q=p+1}^{n} Y_{pq} V_q^r] / Y_{pp} \quad (27)$$

3. A load-flow computation as defined in claim1 is characterized in the use of a method of decomposing a network referred to as Suresh's diakoptics that involves determining a sub-network for each node involving directly connected nodes referred to as level-1 nodes and their directly connected nodes referred to as level-2 nodes and so on, and the level of outward connectivity for local solution of a sub-network around a given node is to be determined experimentally.

4. A load-flow computation as defined in claim-1 is characterized in the use of Parallel solving of all sub-networks defined in claim-3 using available solution estimate at the start of the iteration, initializing a vector of dimension equal to the number of nodes with each element value zero, adding solution estimates for a node resulting from different sub-networks in a corresponding vector element, counting the number of additions and calculating new solution estimate or corrections to the available solution estimate using any relevant relations in the following with the number '3' replaced by the actual number of contributions or additions to a vector element, and storing it as initial available estimate for the next iteration

$$V_p^{(r+1)} = (V_{p1}^{(r+1)} + V_{p2}^{(r+1)} + V_{p3}^{(r+1)})/3 \quad (30)$$

$$\Delta\theta_p^{(r+1)} = (\Delta\theta_{p1}^{(r+1)} + \Delta\theta_{p2}^{(r+1)} + \Delta\theta_{p3}^{(r+1)})/3 \quad (31)$$

$$\Delta V_p^{(r+1)} = (\Delta V_{p1}^{(r+1)} + \Delta V_{p2}^{(r+1)} + \Delta V_{p3}^{(r+1)})/3 \quad (32)$$

$$V_p^{(r+1)} = \sqrt{(\mathrm{Re}((V_{p1}^{(r+1)})^2) + \mathrm{Re}((V_{p2}^{(r+1)})^2) + \mathrm{Re}((V_{p3}^{(r+1)})^2))/3} + j\sqrt{(\mathrm{Im}((V_{p1}^{(r+1)})^2) + \mathrm{Im}((V_{p2}^{(r+1)})^2) + \mathrm{Im}((V_{p3}^{(r+1)})^2))/3} \quad (33)$$

$$\Delta\theta_p^{(r+1)} = \sqrt{(\Delta\theta_{p1}^{(r+1)})^2 + (\Delta\theta_{p2}^{(r+1)})^2 + (\Delta\theta_{p3}^{(r+1)})^2)/3} \quad (34)$$

$$\Delta V_p^{(r+1)} = \sqrt{(\Delta V_{p1}^{(r+1)})^2 + (\Delta V_{p2}^{(r+1)})^2 + (\Delta V_{p3}^{(r+1)})^2)/3} \quad (35)$$

5. A load-flow computation as defined in claim-1. Claim-2, claim-3 and claim-4 is characterized in the use of the simplified parallel computer a server processor-array processors architecture, where each of the array processors communicate only with server processor and commonly shared memory locations and not among themselves. (Fig.4)